

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Serial No. 10/655,326

Confirmation No. 6111

I hereby certify that this correspondence is being transmitted to the United States Patent & Trademark Office via electronic submission or facsimile on the date indicated below:

12/08/2009
Date

/Pamela Gerik/
Pamela Gerik

SUPPLEMENTAL APPEAL BRIEF

Dear Sir:

Further to the Notice of Non-Compliant Appeal Brief mailed November 19, 2009, Appellant presents this Supplemental Appeal Brief. Appellant hereby appeals to the Board of Patent Appeals and Interferences from the final rejection of pending claims 1-4, 6, 7, 11, 13-21, 25, 26, 32, 34, 36, 37, 39, and 42, and respectfully requests that this appeal be considered by the Board.

I. REAL PARTY IN INTEREST

The subject application is owned by Ramal Acquisition Corp., a wholly owned subsidiary of Pervasive Software, Inc., as evidenced by the document recorded at reel 018779 and frame 0519.

II. RELATED APPEALS AND INTERFERENCES

No appeals, interferences, or judicial proceedings are known which would directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF THE CLAIMS

Claims 1-4, 6, 7, 11, 13-15, 25, 26, 32, 34, 36, 37, 39, and 42 stand finally rejected and are the subject of this appeal. Claims 5, 8-10, 12, 16-24, 27-31, 33, 35, 38, 40, 41, 43, and 44 are canceled.

IV. STATUS OF AMENDMENTS

The claims were amended subsequent to their final rejection in order to simplify matters for appeal. The amendment after final was entered in part, but was not entered as to claim 1. The Appendix hereto reflects the current state of claims. Appellants respectfully request entrance of the amendments to claim 1 filed May 8, 2009.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Independent claim 1 recites a storage medium comprising a Software Development Kit (SDK) having program components which are executable through a common application program interface (Specification -- pg. 3, lines 6-9; Figs. 1, 2), wherein the program components comprise: a first program component having coding directives which are utilizable by a developer to write program instructions that are executable by a processor for adaptively navigating through one or more websites (Specification -- pg. 3, lines 16-19); one or more additional program components having coding directives which are utilizable by a developer to write program instructions that are executable by a processor (Specification -- pg. 4, lines 1-8;) for: extracting scripted content, wherein the content is only visible when loaded and executed, from the one or more websites including loading and executing the scripted content (Specification -- pg. 6, lines 1-9); and storing the extracted scripted content at a target location (Specification -- pg. 6, lines 8-14; Fig. 3).

Dependent claim 2 recites the storage medium of claim 1, wherein the coding directives of the first program component are utilizable by a developer to write program instructions that

are executable by a processor for conditionally navigating through the one or more websites (Specification -- pg. 3, lines 19-24).

Dependent claim 3 recites the storage medium of claim 1, wherein the coding directives of the first program component are utilizable by a developer to write program instructions that are executable by a processor for facilitating navigation through the one or more websites (Specification -- pg. 3, lines 16-25).

Dependent claim 4 recites the storage medium of claim 3, wherein the coding directives of the first program component are utilizable by the developer to selectively write the program instructions associated with facilitated navigation for specific timeframes (Specification -- pg. 12, lines 13-16).

Dependent claim 6 recites the storage medium of claim 1, wherein the coding directives of the one or more additional program components are further utilizable by a developer to write program instructions that are executable by a processor for standardizing the scripted and unscripted content (Specification -- pg. 4, lines 19-25; pg. 5, lines 26-30; Fig. 3).

Dependent claim 7 recites the storage medium of claim 1, wherein the coding directives of the one or more additional program components are further utilizable by a developer to write program instructions that are executable by a processor for generating a model of logical structure of the scripted and unscripted content (Specification -- pg. 5, lines 16-20; pg. 8, lines 6-9).

Dependent claim 11 recites the storage medium of claim 1, wherein the coding directives of the first program component are further utilizable by a developer to write program instructions that are executable by a processor for: recognizing a scripting language embedded within the one or more websites (Specification -- pg. 6, lines 1-9; Fig. 3); and executing the embedded scripting language using an interpreter (Specification -- pg. 5, lines 1-9; Fig. 3).

Dependent claim 13 recites the storage medium of claim 1, wherein the coding directives of the first program component are utilizable by a developer to write program instructions that are executable by a processor for accessing data other than what may be configured to be displayed on a browser as characterized by a structural layout of an accessed website (Specification -- pg. 3, lines 25-27).

Dependent claim 14 recites the storage medium of claim 1, wherein the coding directives of the one or more program components are further utilizable by a developer to write program instructions that are executable by a processor for posting data on the one or more websites (Specification -- pg. 4, lines 14-17).

Dependent claim 15 recites the storage medium of claim 1, wherein the coding directives of the first program component and one or more additional program components are utilizable by a developer to write event driven program instructions (Specification -- pg. 4, lines 15-17; pg. 7, lines 9-12; pg. 15, lines 25-28).

Independent claim 25 recites a Software Development Kit (SDK) comprising program instructions executable using a processor (Specification -- pg. 3, lines 6-9; Figs. 1, 2) for: navigating through a website to access information (Specification -- pg. 3, lines 17-22; Fig. 3); parsing the accessed information into a model of logical structure (Specification -- pg. 5, lines 16-20; Fig. 3); loading and executing a scripting language embedded within the website such that results of the script execution can be parsed into the model of logical structure (Specification -- pg. 5, lines 17-20; pg. 6, lines 1-10; pg. 16, lines 1-9; Fig. 3); searching for content, at least some of which is available but not displayed on said website, within the model of logical structure (Specification -- pg. 3, lines 25-27; pg. 5, lines 17-20; pg. 14, lines 19-23, pg. 14, line 30 – pg. 15, line 5); extracting, independent of user intervention, the searched content from the one or more websites (Specification -- pg. 6, lines 21-25; Fig. 3); and storing, independent of user intervention, the extracted content at a target location (Specification -- pg. 6, lines 8-12).

Dependent claim 32 recites the storage medium of claim 25, wherein the program instructions are further for posting data upon the website (Specification -- pg. 4, lines 14-17).

Dependent claim 34 recites the storage medium of claim 25, wherein the program instructions are for monitoring the status of the accessed information on the website, and for sending an alert upon detecting a change in the status of the accessed information (Specification -- pg. 6, lines 13-18).

Independent claim 36 recites a computer-implemented method of using a Software Development Kit for obtaining a collection of information from one or more websites (Specification -- pg. 3, lines 4-9; Figs. 1, 2), comprising: accessing the one or more websites (Specification -- pg. 3 lines 4-9; Fig. 3); partitioning contents on the one or more websites into a model of logical structure (Specification -- pg. 5, lines 17-20); loading and executing a script embedded within the one or more websites such that results of the script execution can be parsed into the model of logical structure (Specification -- pg. 6, lines 4-9); querying the model of logical structure for information of interest (Specification -- pg. 6, lines 21-25); automatically extracting, independent of user intervention, the information of interest from the one or more websites (Specification -- pg. 6, lines 21-25; Fig. 3); automatically storing, independent of user intervention, the extracted information of interest at a target location (Specification -- pg. 6, lines 21-25; Fig. 3); and navigating one or more websites based, in part, on the extracted information of interest (Specification -- pg. 7, lines 5-8).

Dependent claim 37 recites the computer-implemented method of claim 36, further comprising standardizing the contents on the one or more websites into a standard format prior to the step of partitioning (Specification -- pg. 4, lines 19-25; pg. 5, lines 26-30; Fig. 3).

Dependent claim 39 recites the computer-implemented method of claim 36, further comprising posting data upon a website in response to the step of extracting the information of interest from the one or more websites (Specification -- pg. 4, lines 14-17).

The present invention relate to methods and systems to conditionally navigate and extract content, particularly scripted content, from websites without user intervention. "Scripted content" may refer to website content which includes one or more executable scripts from which to access the content (Specification -- pg. 2, lines 16-19). "Unscripted content," may generally

refer to website content which does not depend on client-side scripts in order to obtain the content. In other words, unscripted content may refer to information displayed on a website that is governed by the website's server (Specification -- pg. 2, lines 16-19; pg. 8, lines 6-10).

VI. GROUNDS OF REJECTION TO BE REVIEWED

1. The rejection of claims 1-3, 6, 7, 13, 14, 25-26, 32, 34, 36, 39, and 42 under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,209,007 to Kelley et al. (hereinafter "Kelley") in view of "Effective Web data extraction with standard XML technologies" by Myllymaki (hereinafter "Myllymaki").
2. The rejection of claim 4 under U.S.C. § 103(a) as being unpatentable over Kelley, Myllymaki, and U.S. Patent Application Publication No. 2004/0143567 to Gross et al. (hereinafter "Gross").
3. The rejection of claims 11 and 15 under U.S.C. § 103(a) as being unpatentable over Kelley, Myllymaki, and U.S. Patent No. 7,047,318 to Svedloff (hereinafter "Svedloff").
4. The rejection of claim 37 under U.S.C. § 103(a) as being unpatentable over Kelley and U.S. Patent Application Publication No. 2002/0143821 to Jakubowski.

VII. ARGUMENT

1. **The rejection of claims 1-3, 6, 7, 13, 14, 25-26, 32, 34, 36, 39, and 42 under 35 U.S.C. § 103(a) as being unpatentable over Kelley and Myllymaki.**

The final office action fundamentally misapplies the primary reference of Kelley. The present claimed invention relates to a software development kit for navigating web sites and extracting "scripted" content. Kelley is cited to show that Javascript can be loaded and executed to display content. While this may be true, it has little to do with the claims presented. Myllymaki is cited to show an example of a system for "extracting" hidden content from the

web. Even if properly applied, the combination of Kelley and Myllymaki does not address several limitations in the independent claims on appeal.

The final office action complains that the Dingman declaration and Applicant's arguments fail to "address the fundamental objective of Kelley, which is to view Web content on a display screen." (p. 4) As explained in the present application a primary objective of the present invention is to adaptively navigate websites to extract scripted and unscripted content without user intervention. For example in "some embodiments, the first program component may be used to access information other than what may be configured to be displayed on a browser as characterized by the structural layout of an accessed website." P. 3, lines 25-27. The application describes different methodologies, such as generating a model of logical structure of the website content, parsing accessed information into the model, posting data, and standardizing content. pp. 4-5.

Whether or not Kelley, Internet Explorer, Mozilla Firefox allows Javascript to be executed and an output screen generated is not relevant to the claims presented and discussed herein.

The proper context for determining the issue of obviousness or nonobviousness is the Supreme Court Decision in *Graham v. John Deere*, 383 US 1, 148 USPQ 459 (1966), *KSR International Co. v. Teleflex Inc.*, 127 S. Ct 1727 (2007). See also, MPEP 706 and 2141. The primary considerations are: the scope and content of the prior art; the differences between the prior art and the claims in issue; the level of ordinary skill in the pertinent art; and secondary considerations, such as commercial success, long felt and unresolved needs, failures of others, etc.

The so-called "teaching, suggestion, or motivation" test is not to be rigidly applied before analyzing the *Graham* factors. *KSR International Co. v. Teleflex Inc.*, 127 S. Ct 1727 (2007). However, there still must be an explicit reason that would prompt a person of ordinary skill in the art to combine the elements in the way the claimed invention does. *KSR*, 127 S. Ct. at 1731. See also, *Takeda Chemical Industries, LTD v. Alphapharm PTY*, 06-1329 (Fed. Cir. 2007) slip

op. at 10. Of course, if the proposed modification or combination would change the principle of operation, then the references are not sufficient to render the claims *prima facie* obvious. *In re Ratti*, 270 F.2d 810, 123 U.S.P.Q. 349 (CCPA 1959).

Even after *KSR International Co.* there must be some suggestion in the record for the proposed modification. See *Takeda Chemical Industries, LTD slip op. at 10*, see also e.g. PTO guidance letter regarding KSR. “Although predictability is a touchstone of obviousness, the “predictable result” discussed in KSR refers not only to the expectation that prior art elements are capable of being physically combined, but also that the combination would have worked for its intended purpose.” *Depuy Spine v. Medtronic*, 2008-1240, -1253, -1401 (Fed. Cir. 2009) slip op. at 13 (citing *KSR International Co.*).

A. Discussion of the Prior Art

Kelley relates to a system for creating a customized web page containing information from other web pages (Kelley -- Col. 2, lines 4-8). Kelley simply scans HTML code and applies Boolean logic to create a new web page (Kelley -- Col. 4, lines 58-65). Kelley does “extract” Javascript code (Kelley -- Col. 7, lines 19-22) and attempts to assemble the extracted source. However, the extracted Javascript code is used to build new code.

Kelley does not relate to software development kits or teach the processes described and claimed in the present application. For example, claim 1 call for “adaptively navigating” through websites. Examples of such adaptive navigation are given in the present specification, e.g. at page 13. Kelley does not teach this “adaptive navigation” or “conditional navigation.”

Further, Kelley does not teach “extracting scripted content” where the scripted content is “loaded and executed” as more fully described in the specification at page 23 and in Figure 3. As described, this “extract content” allows access to content that is only visible when scripts are loaded and executed by a client. In contrast, Kelley is cited at Figure 5 as “extracting Javascript code” at 340. As can be seen from Figure 3 and Col. 6 of Kelley, Kelley is really just associating hyperlinks in the page as levels. Reading Kelley, at Col. 4, lines 1-9, and Col. 6, lines 1-8, the

Javascript is placed in the HTML to build an output screen. At Col. 7, lines 16-25, of Kelley, the javascript is used to generate the output screen. This is in marked contrast to the present invention as claimed.

The Office Action of 12/29/06 characterizes Kelley as follows:

Kelley does not expressly disclose: wherein the one or more additional program components comprise a third program component with coding directives which are utilizable by the developer to write program instructions for searching for information within the model of logical structure (Office Action at pg. 14).

Myllymaki does recognize a problem addressed by one aspect of the present invention – namely navigating and extracting hidden and inaccessible data on the Web. However, Myllymaki uses a conventional crawler/spider approach of analyzing HTML forms and JavaScript Code using filters to form a list of static hyperlinks. (p. 640, Col. 1). While this approach may be interesting, it does not relate to the claimed approach of the present invention.

B. Discussion of the Claims

i. Claim 1

Claim 1 specifies the coding directives of the first program component are utilizable by a developer to write processor-executable program instructions for adaptively navigating through the one or more websites. There is no teaching or suggestion within Kelley that the Boolean search taught therein is configured with parameters to avoid or overcome obstacles which may be encountered along a navigational route. Therefore, Kelley fails to disclose adaptive navigation through one or more websites as specified in claim 1 and defined on page 13 of the specification; nor can Kelley be combined with Myllymaki to arrive at the limitations of claim 1.

Claim 1 further calls for “extracting scripted content,” where some of the scripted content is available, but not displayed (or visible) including loading and executing scripted content. Kelley does not address this limitation. The final office action posits that Fig. 5, element 350, of

Kelley meets this limitation -- Appellants respectfully disagree. Kelley appears to extract javascript source in an attempt to assemble new source (Kelley -- Col. 7, lines 19-25).

ii. Claim 2

Claim 2 specifies the coding directives of the first program component are utilizable by a developer to write processor-executable program instructions for conditionally navigating through the one or more websites. Kelley teaches the use of a Boolean search to search websites, but there is no teaching or suggestion within Kelley that such a Boolean search includes searching websites with if, then and looping directives. Therefore, Kelley fails to disclose conditionally navigating through one or more websites as specified in claim 2 and defined on page 12, lines 5-13, of the specification; nor can Kelley be combined with Myllymaki to arrive at the limitations of claim 2.

iii. Claim 3

Claim 3 specifies the coding directives of the first program component are utilizable by a developer to write processor-executable program instructions for facilitating navigation through the one or more websites. There is no teaching or suggestion within Kelley that the Boolean search taught in Kelley is configured with parameters to avoid or overcome obstacles which may be encountered along a navigational route. Therefore, Kelley fails to disclose facilitating navigation through one or more websites as specified in claim 3 and defined on page 13 of the specification; nor can Kelley be combined with Myllymaki to arrive at the limitations of claim 3.

iv. Claim 6

Claim 6 specifies the coding directives are utilizable by the developer for writing processor-executable program instructions with which to standardize spaces within the web page content. Kelley's use of a Boolean search to search for spaces does not meet or suggest this limitation. As noted on page 17, lines 21-23 of the specification, the term "standardizing" in the presently claimed case refers to modifying the arrangement of web content into a

comprehensible and standard format (underline added for emphasis). A teaching of searching for spaces within a web page does not lend to modifying the arrangement of web content and, therefore, does not teach the limitations of claim 6.

v. Claim 7

Claim 7 is directed to a second program component with coding directives which are utilizable by the developer to write program instructions for generating a model of logical structure of the scripted and unscripted content. As explained in the specification (p. 16, lines 1-12) a model of logical structure defines the logical structure of a document and the way the document is accessed and manipulated. One model currently available is the Document Object Model (DOM) authored by the World Wide Web Consortium (W3C). The DOM specification is available on the Web at URL <http://www.w3.org/>. Regardless of the model used, parsing component 15 may, in part, be used to extract scripted and/or unscripted content from the one or more websites. None of the references cited address this element.

vi. Claim 13

Claim 13 specifies the coding directives of the first program component are utilizable by a developer to write processor-executable program instructions for accessing data other than what may be configured to be displayed on a browser as characterized by a structural layout of an accessed website. The Examiner cites column 4, lines 2-7, of Kelley as teaching that javascript code on a web browser may be accessed. Javascript code is configured to be displayed on web browsers and, therefore, such a passage cannot be used to reject the limitations of claim 13. Thus, Kelley does not teach the limitations of claim 13; nor can Kelley be combined with Myllymaki to arrive at the limitations of claim 13.

vii. Claim 14, 32, and 39

Claims 14, 32, and 39 specify processor-executable program instructions for posting data on the one or more websites that were navigated for the information of interest. Kelley teaches

creating a separate customized web page with information extracted from other web pages. However, there is no teaching or suggestion within Kelley of posting the extracted data on the web pages that are searched and, therefore, does not teach or suggest the limitations of claims 14, 32, and 39. It is noted that the term “posting” in the present claims refers to the HTTP POST request method, a process which is commonly used by browsers for sending data submitted on a form of a webpage for processing. It is asserted that one skilled in the industry of web navigation and extraction is aware that the term “posting” generally refers to the HTTP POST request method and would be apprised that the scope of the captioned application includes such a process by use of the term “posting.” The distinction between the limitations of present claims and web browsers employing the HTTP POST method is that the processor-executable program instructions of the present claims are configured to mimic a webpage browser to build a HTTP POST request. Thus, Kelley does not teach the limitations of claim 14, 32, and 39; nor can Kelley be combined with Myllymaki to arrive at the limitations of claim 14, 32, and 39.

viii. Claims 25, 36

Independent claim 25 recites in part: “parsing the accessed information into a model of logical structure” and “loading and executing a scripting language embedded within the website such that results of the script execution can be parsed into the model of logical structure” Independent claim 36 includes a similar limitation for a computer-implemented method for obtaining a collection of information from one or more websites. As noted above, there is no teaching or suggestion within Kelley that code on a website is executed to retrieve information of interest. Rather, Kelley appears to teach extracting code from a website and assembling the code to build a new code for generating information of interest or references to information of interest (Kelley -- Col. 7, lines 19-25).

Further, as explained in the specification (p. 16, lines 1- 12) a model of logical structure defines the logical structure of a document and the way the document is accessed and manipulated. One model currently available is the Document Object Model (DOM) authored by the World Wide Web Consortium (W3C). The DOM specification is available on the Web at URL <http://www.w3.org/>. Regardless of the model used, parsing component 15 may, in part, be

used to extract scripted and/or unscripted content from the one or more websites. None of the references cited address this element. The Final Office Action cites Kelley, at Fig. 5, element 350 and Col. 7, lines 16 – 25 as meeting or suggesting this limitation – “parsed into the model of logical structure.” However, nowhere in the cited passages does Kelley suggest such an action. This element is not found or suggested in any of the art cited.

Consequently, independent claims 25 and 36 are asserted to be patentably distinct from Kelley and Myllymaki.

ix. Claim 34

Claim 34 calls for sending “an alert upon detecting a change in the status of the accessed information.” There is no suggestion in Kelley, nor can Kelley be combined with Myllymaki, to teach at this limitation.

2. The rejection of claim 4 under U.S.C. § 103(a) as being unpatentable over Kelley, Myllymaki, and Gross.

Claim 4 specifies the coding directives of the first program component are utilizable by a developer to selectively write the program instructions associated with facilitated navigation for specific timeframes (Specification -- pg. 13, lines 23-27). There is no teaching or suggestion within Kelley of modifying the manner in which websites are navigated, much less with respect to the timeframes they are navigated. Consequently, Kelley fails to teach or suggest, nor can Kelley be combined with Myllymaki and Gross to teach or suggest the limitations of claim 4.

3. The rejection of claims 11 and 15 under U.S.C. § 103(a) as being unpatentable over Kelley, Myllymaki, and Svedloff.

i. Claim 11

Claim 11 specifies the storage medium of claim 1 includes an interpreter for executing embedded scripting languages. The Examiner cites Kelley as teaching the process of searching web pages for the existence of Java code or other language code to reject the limitations of claim

11. As noted on page 16, lines 21-22 of the specification, the term “interpreter” of the presently claimed case refers to a set of programming instructions which is configured to execute scripting languages. Since Kelley does not teach the execution of scripting languages to access scripted content, there is no teaching or suggestion to include an interpreter. Further, there is no teaching or suggestion that code is executed to build new code (*See*, element 350 of Fig. 5 in Kelley, cited by the Examiner), nor does column 7, lines 23-25 (which was also cited by the Examiner) teach or suggest executing embedded scripting language to access scripted content. Instead, such a passage teaches taking extracted javascript code to build the new code.

ii. Claim 15

Claim 15 addresses using the coding directives to write event driven program instructions. As explained in the specification at p. 15, a developer may create program instructions which navigate and extract content from a website upon detecting a change in content or the location of content upon a particular website – i.e. an “event.” Such program instructions may be configured to be automatic (i.e., implemented without intervention from a user of the program instructions) or may be invoked by command specified in the program instructions. In any case, the event driven program instructions may be associated with any occurrence in which the content or location of content on a website has changed, such as a refresh or a redirection of the website.

The final office action (page 19) acknowledges that Kelley does not teach, this limitation, but asserts that this limitation is taught by Svedloff. However, the cited passage in Svedloff apparently relates to using a Java Server Pages for interactive web pages. Even if true, this combination does not address the limitation as claimed.

4. The rejection of claim 37 under U.S.C. § 103(a) as being unpatentable over Kelley and Jakubowski.

Claim 37 specifies the coding directives are utilizable by the developer for writing processor-executable program instructions with which to standardize spaces within the web page content. Kelley’s use of a Boolean search to search for spaces does not meet or suggest this

limitation. As noted on page 17, lines 21-23 of the specification, the term “standardizing” in the presently claimed case refers to modifying the arrangement of web content into a comprehensible and standard format (underline added for emphasis). A teaching of searching for spaces within a web page does not lend to modifying the arrangement of web content and, therefore, does not teach the limitations of claim 37.

* * *

In conclusion, the claims currently presented are allowable over the § 103 issues raised in the final Office Action, and applicant respectfully requests reconsideration and allowance in view of the traversal herein.

Respectfully Submitted,

/Charles D. Huston/

Charles D. Huston
Registration No. 31,027
Attorney for Appellant

Customer No. 35617
Date: July 22, 2009

VIII. APPENDIX

The present claims on appeal are as follows.

1. A storage medium comprising a Software Development Kit (SDK) having program components which are executable through a common application program interface, wherein the program components comprise:

a first program component having coding directives which are utilizable by a developer to write program instructions that are executable by a processor for adaptively navigating through one or more websites;

one or more additional program components having coding directives which are utilizable by a developer to write program instructions that are executable by a processor for:

extracting scripted content, wherein the content is only visible when loaded and executed, from the one or more websites including loading and executing the scripted content; and

storing the extracted scripted content at a target location.

2. The storage medium of claim 1, wherein the coding directives of the first program component are utilizable by a developer to write program instructions that are executable by a processor for conditionally navigating through the one or more websites.

3. The storage medium of claim 1, wherein the coding directives of the first program component are utilizable by a developer to write program instructions that are executable by a processor for facilitating navigation through the one or more websites.

4. The storage medium of claim 3, wherein the coding directives of the first program component are utilizable by the developer to selectively write the program instructions associated with facilitated navigation for specific timeframes.

6. The storage medium of claim 1, wherein the coding directives of the one or more additional program components are further utilizable by a developer to write program instructions that are executable by a processor for standardizing the scripted and unscripted content.

7. The storage medium of claim 1, wherein the coding directives of the one or more additional program components are further utilizable by a developer to write program instructions that are executable by a processor for generating a model of logical structure of the scripted and unscripted content.

11. The storage medium of claim 1, wherein the coding directives of the first program component are further utilizable by a developer to write program instructions that are executable by a processor for:

recognizing a scripting language embedded within the one or more websites; and

executing the embedded scripting language using an interpreter.

13. The storage medium of claim 1, wherein the coding directives of the first program component are utilizable by a developer to write program instructions that are executable by a processor for accessing data other than what may be configured to be displayed on a browser as characterized by a structural layout of an accessed website.

14. The storage medium of claim 1, wherein the coding directives of the one or more program components are further utilizable by a developer to write program instructions that are executable by a processor for posting data on the one or more websites.

15. The storage medium of claim 1, wherein the coding directives of the first program component and one or more additional program components are utilizable by a developer to write event driven program instructions.

25. A Software Development Kit (SDK) comprising program instructions executable using a processor for:

navigating through a website to access information;

parsing the accessed information into a model of logical structure;

loading and executing a scripting language embedded within the website such that results of the script execution can be parsed into the model of logical structure;

searching for content, at least some of which is available but not displayed on said website, within the model of logical structure;

extracting, independent of user intervention, the searched content from the one or more websites; and

storing, independent of user intervention, the extracted content at a target location.

26. The storage medium of claim 25, wherein the program instructions are further for accessing the website without a user interface.

32. The storage medium of claim 25, wherein the program instructions are further for posting data upon the website.

34. The storage medium of claim 25, wherein the program instructions are for monitoring the status of the accessed information on the website, and for sending an alert upon detecting a change in the status of the accessed information.

36. A computer-implemented method of using a Software Development Kit for obtaining a collection of information from one or more websites, comprising:

accessing the one or more websites;

partitioning contents on the one or more websites into a model of logical structure;

loading and executing a script embedded within the one or more websites such that results of the script execution can be parsed into the model of logical structure;

querying the model of logical structure for information of interest;

automatically extracting, independent of user intervention, the information of interest from the one or more websites;

automatically storing, independent of user intervention, the extracted information of interest at a target location; and

navigating one or more websites based, in part, on the extracted information of interest.

37. The computer-implemented method of claim 36, further comprising standardizing the contents on the one or more websites into a standard format prior to the step of partitioning.

39. The computer-implemented method of claim 36, further comprising posting data upon a website in response to the step of extracting the information of interest from the one or more websites.

42. The computer-implemented method of claim 36, including monitoring the status of the contents on the one or more websites, and further comprising performing the steps of partitioning, querying, automatically extracting, and automatically storing upon detecting a change in the status of the contents on the one or more websites.

IX. EVIDENCE APPENDIX

A declaration of Paul Dingman under 37 C.F.R. § 1.132 regarding the teachings of Kelley was filed August 26, 2008.

X. RELATED PROCEEDINGS APPENDIX

No prior or pending appeals, interferences, or judicial proceedings are known to Appellant or Assignee which would directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.